Multistage Ring Network: A New Multiple Ring Network for Large Scale Multiprocessors *

Dongho Yoo, Inbum Jung and Seung Ryoul Maeng
Department of Computer Science, Korea Advanced Institute of Science and Technology, Korea
e-mail: {dhyoo, jib, maeng}@camars.kaist.ac.kr

Abstract

We present a new multiple ring network for multiprocessors, called the Multistage Ring Network(MRN). The MRN has a 2-level hierarchy of register insertion rings, and its interconnection of global rings forms a type of the multistage network. The architecture of the MRN is effective at diffusing the global traffic to all global rings and the bandwidth of the network increases proportionally with increases in the system size. At the same time, the MRN retains the economic design and high speed communication advantages of the ring network. We develop a deadlock-free routing algorithm for the MRN. We also present a performance analysis of the MRN and compare the results with those of a hierarchical ring network.

1. Introduction

Bit parallel and unidirectional ring-based connections have been found to be a very promising interconnection technology for multiprocessors due to their simple hardware interfaces, high speed communication, wider data path, and easy addition of extra nodes. The potential of the ring connections is demonstrated by the SCI (Scalable Coherent Interface IEEE standard 1596)[5], which shows a very high transfer rate, up to 1 Gigabyte per second per link.

To accommodate a large number of processors, multiple rings need to be interconnected because a single ring does not scale well due to the fixed bandwidth of the ring, independent of ring size. Rings are connected to each other by *switches* having multiple input links and output links. Because of ring structure, each of the input links of the switch has a unique preferred output link, and routing to the preferred output link is significantly faster than routing to the other links [8]. Therefore, rings must be interconnected to minimize the number of ring changes. Another factor determining the performance of the system is the complexity of the switch. The switch having a wide fanout may reduce the distance between a source node and a destination node, but it makes the board logic too complex. The point-to-point connection is so fast that the complex board logic causes a performance bottleneck. It is also important to keep each ring size within certain ranges [11] because a large ring tends to saturate rapidly when the load on the ring increases. In addition, the routing algorithm for a topology with multiple rings should be designed carefully to avoid deadlock that may occur.

Much research has been done on the interconnection of multiple rings. Some classical topologies such as Butterfly and k-ary n-cube were synthesized with rings in [8] and [12]. These approaches require complex hardwares and have no considerations for the localized traffic. On the other hand, the hierarchical ring networks have been found to be cost-effective interconnection

*This study is partially supported by Samsung Electronics Co., Ltd

networks [6], [9], [7], [1]. But this topology has limited scalability because the bandwidth of the network decreases as one moves toward the top of the hierarchy.

In this paper we present the design of a new interconnection scheme, called the Multistage Ring Network (MRN), for multiprocessors with up to a few thousands processors. The MRN has a two level hierarchy with register insertion rings. A lowest-level ring, local ring, consists of processing modules, and each local ring is connected to two rings in the top level. The rings in the top level, global rings, are connected with each other to form a type of the multistage network. Having this network structure, the MRN can isolate the localized traffic from the global rings and diffuse the traffic at the top level effectively. The switch of the MRN has three input links and three output links. We show a prototype design of the switch that can be implemented with simple logic gates. We also show a deadlock-free self-routing algorithm for the MRN.

The remainder of this paper is organized as follows; In the following section we describe in more detail the architecture of the MRN. In Section 3 we describe the simulation experiments. The results and comparisons with those of a hierarchical ring network are reported in Section 4. Section 5 concludes our work.

2. Multistage Ring Network

2.1. Structure and Operation

An MRN(r, l) will, in general, consist of $N = r2^r l$ processing modules, where r is the number of stages and l is the number of nodes per local ring. A local ring is connected to two global rings, of which one is called *horizontal ring* and the other is *diagonal ring*. The top level structure is constructed as follows. Let (α_i, α_j) be an identifier of a local ring at stage α_i and row α_j on the top level (refer Figure 1). A horizontal ring connects local rings in the same row and also has connections with r diagonal rings. A diagonal ring connects the series of local rings,

$$(1, \alpha_1), (2, \alpha_2), \cdots, (r, \alpha_r), (1, \alpha_{r+1}),$$

$$(2, \alpha_{r+2}), \cdots, (r, \alpha_{2r})$$

where $\alpha_{j+1} = \alpha_j \text{ XOR } (2^{r-1}/2^{((j-1) \bmod r)})$ for $1 \leq j$ and $0 \leq \alpha_1 < 2^{r-1}$. Each diagonal ring is connected to $2r$ horizontal rings. Thus, the size of a diagonal ring is two times larger than that of a horizontal ring.

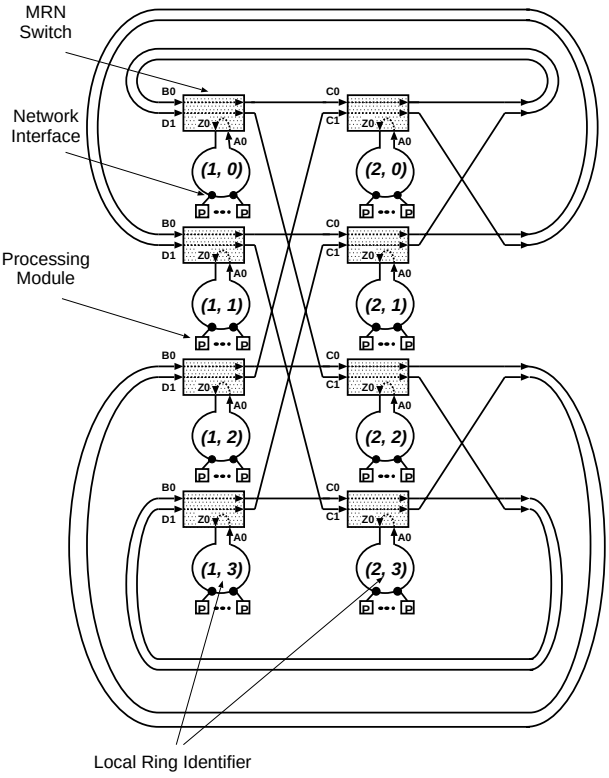

Figure 1. An example of the MRN with eight local rings and two stages. Each dotted line in the MRN switches denotes the preferred connection of a given input link. The letters demonstrate deadlock free routing, described in the text.

Two types of interfaces are used to construct the MRN: a *network-interface* connects a processing module to a local ring, and an *MRN switch* connects a local ring to two global rings. The network-interface controls outgoing packets from the processing module and switches incoming

packets from the input link to the processing module or the output link according to their routing information. The MRN switch controls the packet transfers between the three rings.

Figure 2 shows the general structure of the network-interface in the register insertion ring architecture. The *receive queue* stores incoming packets destined to the processing module, and the *transmit queue* stores the packets that will be injected into the ring. Both queues are split into response and request queues to prevent possible deadlocks [5]. The *ring buffer* stores packets going to the downstream nodes while the output link is transmitting another packet in the transmit queue or the ring buffer. When the ring buffer is empty and no packets are being transmitted, the incoming packet can be transferred to the output link directly, bypassing the ring buffer. Note that the packets in the transmit queue can have chance to be transmitted to the output link only when the ring buffer is empty.

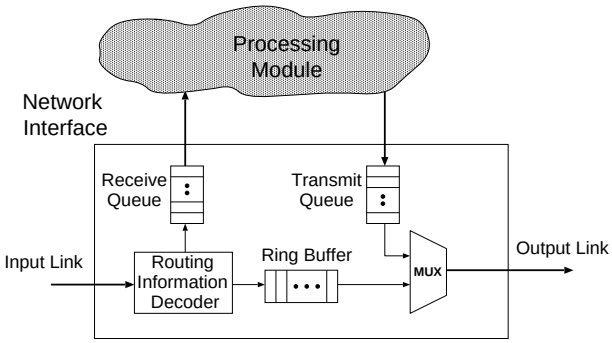

Figure 2. Block diagram of a network-interface.

Figure 3 shows the block diagram of the MRN switch, which can be implemented with simple logic gates. The MRN switch consists of three modules such as *local ring connection module*, *horizontal ring connection module*, and *diagonal ring connection module*. Each module behaves as a node in the given ring. For example, the local ring connection module performs the same operations as a network-interface in the ring. Each module also controls packet transfers

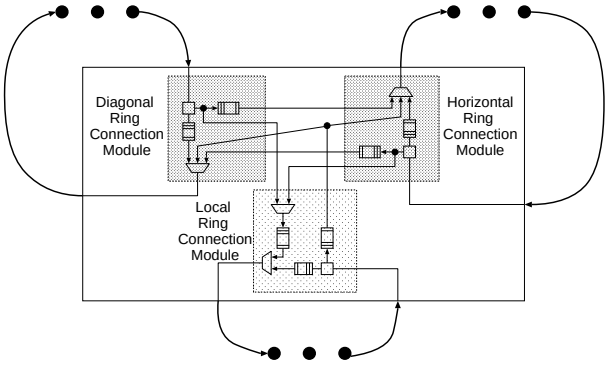

Figure 3. Block diagram of an MRN switch.

between itself and the other modules. In the local ring connection module, the *receive queue* can send its packets directly to the output links of the other two rings, and the *transmit queue* can receive packets directly from the input links of the other two rings. This is an important feature of the MRN switch, to avoid possible deadlocks, which will be described in next subsection. In order to improve performance, the horizontal ring connection module can send packets from the input link to the transmit queue of the local ring connection module while the packets in its receive queue are going to the diagonal ring. The diagonal ring connection module can also accomplish these operations. The queues of the MRN switch are also split into response and request queues.

Processing modules communicate with each other by sending a packet to the target processing module. The target processing module returns the echo packet to the sender after receiving the packet. A packet on the network is sent as a continuous sequence of flits, where the header flits contain routing and flow control information. As flits are forwarded in a bit parallel format, the packet becomes spread across contiguous links of the network. The sending packet moves around the rings until it reaches its target node. In a ring, when the sending packet arrives at the target node or the MRN switch which it must change rings at, the receiving node strips the packet from the ring and stores the packet in its queue. The MRN

switch having received the packet in its queue behaves as a sending node in the ring that the packet should be transferred into.

2.2. Routing Algorithm

The classical deadlock problem may occur in multiple rings. A single ring network has a simple routing strategy in which sending packets are only forwarded until they reach their destinations through the unidirectional path. However, in multiple rings, the entire sending packets are stored in the queues of the switches where they change rings. This causes the same deadlock problem of store-and-forward networks. In the MRN, if packets can start at an arbitrary stage and end at an arbitrary stage, then there exists at least one cycle of queues.

In order to prevent deadlock we use a restrictive routing algorithm and some features of the MRN switch. A packet from a local ring is delivered to its target local ring by the following three routing phases. 1) The packet is transferred to the first stage node via the horizontal ring links. 2) the packet is routed to the horizontal ring containing the target local ring, using horizontal or diagonal ring links as needed. The routing path is determined according to the row address of the destination local ring using the similar manner of determining routing path on the butterfly network [3]. 3) the packet is delivered to the target local ring via the horizontal ring links. Any of the above routing phases can be skipped if it makes no progress toward delivery of the packet.

In Figure 1 we named every queue of the MRN switches to show an example of the proof that the above routing algorithm is deadlock-free. Note that the queues of the MRN switches in the first stage are named differently. This is because the receive queues of the diagonal ring connection modules in the first stage are visited only by the packets being routed in the second routing phase.

Lemma 1 *The routing algorithm for the MRN is deadlock-free.*

Proof. To start the routing in the top level, the packets are stored in the receive queues of the local ring connection modules (named the smallest alphabetically), and they require queues in intra-stage MRN switches. Because the packets are routed along the unidirectional path, they visit queues in alphabetically increasing order in the second routing phase. The packets being routed by the third routing phase require only the transmit queues of the local ring connection modules, named the largest alphabetically. Thus, the packets visit queues in alphabetically increasing order. ■

3. Performance Evaluation

To study the performance of the MRN under practical traffic conditions, we use a simulator because we are unable to find analytical solutions considering the operations after collisions. In this section we describe the simulation, workload parameters.

3.1. Simulator and System Parameters

We constructed a simulator operating on a cycle-by-cycle basis to reflect the behavior of the packets on the network, written in C. The batch mean method was used for output analysis, where the first batch was discarded due to its possible initialization bias. We computed sample means for each batch and computed grand mean and confidence interval using those sample means. Each batch was terminated after running 100000 clock-cycles. As a primary measure of performance *transfer latency* is defined as the time from when a packet is first issued by a processing module until the target node receives the whole packet, measured in network cycle. Each ring network cycle time was assumed to be twice as long as a processor cycle time, values used in recent studies [7]. We assumed that there exists at most one outstanding transaction, and each partition of the queues in the interfaces has enough space to accommodate a packet of the largest size. The SCI

global fairness protocol [10] was used to guarantee approximately equal opportunities of ring access for all nodes in a ring.

There are five main types of packets: read request, read response, write request, write response and echo packets. Only the read response and the write request packets have data, or a cache line, and the write response packet only informs that the requested memory operation has been performed successfully. We assume that the packet with data requires 40 flits, the packet without data requires eight flits, and the echo packet requires four flits, chosen from the SCI packets [5] for its popularity. In a ring the sending packet is striped from the ring by the receiving node, changed to an echo packet with a *Positive/Negative* acknowledge flag and then returned back to its source node. If the source node receives the echo with *Negative* acknowledgment, then it resends the original packet to its destination. When a packet changes rings, the MRN switch having received the packet in its queue behaves as a source node in the ring that the packet should go into.

3.2. Workload Parameters

We use a synthetic workload model characterized by the mean time between cache misses or the inverse of *request rate* λ, the probability of read/write cache miss, and the communication locality. A read miss is assumed to be 0.7 in all experiments, chosen for its consistency with empirical statistics [4]. Our workload model does not consider the cache coherence traffic. We adopt the clusters of locality model by Holliday and Stumm [7] because it has been shown to be effective in many studies of direct networks. The locality model is defined by the number of clusters, each cluster size, and the probability of a requesting packet's target being in each cluster. Consider a system having N processing modules, then $S = (S_1, S_2, S_3)$ and $P = (P_1, P_2, P_3)$ designates that there exists three clusters, ith cluster has S_i closest processing modules for $i \leq 3$, $S_1 + S_2 + S_3 = N$, first cluster has probability P_1 of being the target, and cluster S_i $(1 < i)$ has probability P_i of being the target given that the target is not in any cluster S_k for $k < i$.

4. Results and Comparisons

Because our major interest lies in the performance evaluation of large systems with a few thousands processors within the context of shared memory multiprocessors, the systems considered in the simulation experiments have 1024 processing modules but for the $MRN(5, 8)$, which has system size of 1280 because there exists no ideal number of stages for system size 1024. For the hierarchical ring network, we specify its topology by the branching factor at each level of the hierarchy starting at level 1 ring and ending at the root ring. We consider two topologies, $T_{H8} = (8, 8, 4, 2, 2)$ and $T_{H16} = (16, 4, 4, 2, 2)$, the best topologies in the study of Hollyday [7]. For the request rate, we consider a rate of 0.002 to 0.04 data cache misses per processor cycle; this range is supported by the observed characteristics of a number of application programs in recent studies having shown a mean number of processor cycles between 6 and 137 for shared data accesses [2]. The confidence interval halfwidths for all simulation results reported in this section are 3% or less at a 95% confidence level, except near saturation points having more than a few percent confidence interval halfwidth.

Table 1. Comparing the connection complexities of the systems having system size of 1024 and local ring size of 16.

Metric	Topology	
	$MRN(4, 16)$	$T_{H16} = (16, 4, 4, 2, 2)$
Num. of links	1216	1196
Num. of rings	88	87
Num. of switches	64	86

Figure 4 and Figure 5 show how average latency varies with the request rate λ, and the cluster 1 probability P_1 for the considered systems. In the case of the hierarchical ring network, as the request rate increases from 0.002 to 0.01, the aver-

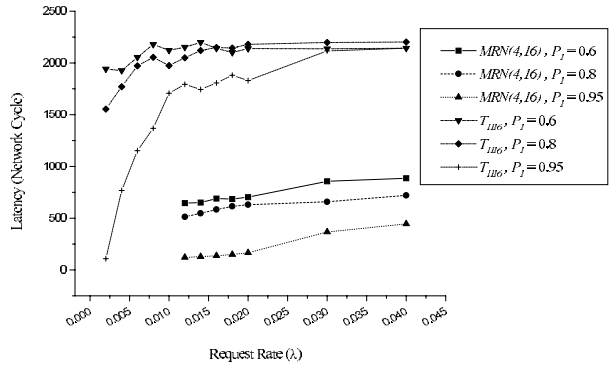

Figure 4. Comparing the average latency of T_{H16} having system size 1024 and $MRN(4, 16)$ for several different cluster 1 probabilities, $P_1 = 0.6, 0.8, 0.95$ when $P_2 = 0.8$, and $S = (1, 4, 1019)$.

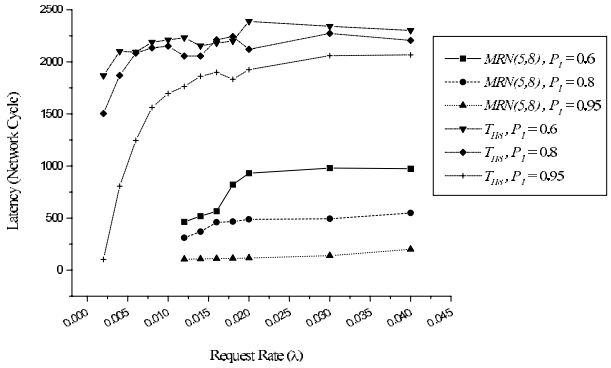

Figure 5. Comparing the average latency of T_{H8} having system size 1024 and $MRN(5, 8)$ for varying P_1 when $P_2 = 0.8$, $S = (1, 4, 1019)$ for T_{H8} and $S = (1, 4, 1275)$ for $MRN(5, 8)$.

age latency increases sharply even for $P_1 = 0.95$. At this point, the average latency for $P_1 = 0.95$ starts to saturate because the network utilization reaches to a peak point and thus most processing units are blocked, waiting the response. The local ring size has minimal effect on the average latency due to the concentrated traffic near the root. In the case of the MRN, the average latency for varying P_1 grows slowly as the request rate increases due to its semi-balanced traffic at global rings. $MRN(5, 8)$ shows better performance than $MRN(4, 16)$, but the difference is negligible for the system loads considered.

Table 1 compares the connection complexities of the hierarchical ring network and the MRN for the identical system and local ring sizes. The detailed hardware complexity comparison for the switches used to construct each system is beyond the scope of this paper because of differences in implementation techniques. Even if the logic complexity of the MRN switch is roughly double that of the switch for the hierarchical ring network, the MRN is cost-effective considering the number of switches used.

5. Conclusion

This paper has presented the design and performance analysis of a Multistage Ring Network(MRN). An efficient deadlock-free routing algorithm for the MRN was also developed. The architecture of the MRN is effective at diffusing the global traffic on the network to all global rings, and the bandwidth of the network increases proportionally as the system size increases. The MRN switch can be implemented with simple logic gates and thus has the potential to scale with the improved circuit technology in the future. Thus, the MRN can retain the advantages of the ring-based design without being subject to scalability limitations. The performance of the MRN is several times better than that of the hierarchical ring network and at comparable cost.

References

[1] D. Basak and D. K. Panda, "Designing large hierarchical multiprocessor systems under processor, interconnection, and packaging advancements," in *Int. Conference on Parallel Processing*, vol. I, pp. 63–66, 1994.

[2] B. Boothe and A. Ranade, "Improved multithreaded techniques for hiding communication latency in multiprocessors," in *Proc. 18th Annual Int. Symp. Comput. Architecture*, (Gold Coast, Australia), pp. 214–223, 1992.

[3] T. Feng, "A survey of interconnection networks," *Computer*, pp. 12–27, Dec. 1981.

[4] K. Gharachorloo, A. Gupta, and J. Hennessy, "Hiding memory latency using dynamic scheduling in shared-memory multiprocessors," in *Proc. 18th Annu. Int. Symp. Computer Architecture*, (Gold Coast, Australia), pp. 22–35, May 1992.

[5] D. B. Gustavson, "The scalable coherent interface and related standards projects," *IEEE Micro*, vol. 12, pp. 10–22, Feb. 1992.

[6] V. C. Hamacher and H. Jiang, "Comparison of mesh and hierarchical networks for multiprocessors," in *ICPP*, vol. I, pp. 67–71, 1994.

[7] M. Holliday and M. Stumm, "Performance evaluation of hierarchical ring-based shared memory multiprocessors," *IEEE Tr. on Computers*, vol. 43, pp. 52–67, Jan. 1994.

[8] R. E. Johnson and J. R. Goodman, "Synthesizing general topologies from ring," in *ICPP*, vol. I, pp. 86–95, 1992.

[9] C. Lam, H. Jiang, and V. C. Hamacher, "Design and analysis of hierarchical ring networks for multiprocessors," in *ICPP*, vol. I, pp. 46–50, 1995.

[10] D. Picker and R. D. Fellman, "An extension to the sci flow control protocol for increased network efficiency," *IEEE/ACM Trans. on Networking*, vol. 4, pp. 71–85, Feb. 1996.

[11] G. Ravindran and M. Stumm, "Hierarchical ring topologies and the effect of their bisection bandwidth constraints," in *ICPP*, vol. I, pp. 51–54, 1995.

[12] B. J. Weding and H. T. Ivar, "Various interconnects for sci-based systems," in *Proc. of Open Bus Systems'91*, (Paris, France), 1991.